Para cambiar la posición de la ventana en ImGui en C, puedes utilizar la función `ImGui::SetNextWindowPos()` antes de comenzar a dibujar la ventana. Aquí tienes un ejemplo básico de cómo puedes hacerlo:

```c
#include "imgui.h"
#include "imgui_impl_sdl.h"
#include "imgui_impl_opengl3.h"

int main()
{
    // Inicializar ImGui y crear una ventana
    ImGui::CreateContext();
    ImGuiIO& io = ImGui::GetIO(); (void)io;

    // Inicializar ImGui para SDL y OpenGL
    ImGui_ImplSDL2_InitForOpenGL(window, gl_context);
    ImGui_ImplOpenGL3_Init(glsl_version);

    // Loop principal
    while (!quit)
    {
        // Iniciar un nuevo frame de ImGui
        ImGui_ImplOpenGL3_NewFrame();
        ImGui_ImplSDL2_NewFrame(window);
        ImGui::NewFrame();

        // Establecer la posición de la próxima ventana
        ImGui::SetNextWindowPos(ImVec2(100, 100), ImGuiCond_FirstUseEver);

        // Comenzar a dibujar la ventana
        ImGui::Begin("Mi Ventana");

        // Contenido de la ventana

        // Finalizar la ventana
        ImGui::End();

        // Renderizar los elementos de ImGui
        ImGui::Render();
        ImGui_ImplOpenGL3_RenderDrawData(ImGui::GetDrawData());

        // Actualizar la ventana y manejar eventos
        SDL_GL_SwapWindow(window);
        SDL_Event event;
        while (SDL_PollEvent(&event))
        {
            ImGui_ImplSDL2_ProcessEvent(&event);
            if (event.type == SDL_QUIT)
                quit = true;
        }
    }
```

```
    // Limpiar ImGui al finalizar
    ImGui_ImplOpenGL3_Shutdown();
    ImGui_ImplSDL2_Shutdown();
    ImGui::DestroyContext();

    return 0;
}
```

En este ejemplo, `ImGui::SetNextWindowPos()` se utiliza para establecer la posición de la próxima ventana en las coordenadas (100, 100). Puedes ajustar estas coordenadas según tus necesidades. Recuerda que este código es solo un ejemplo básico y puede necesitar modificaciones adicionales según tu entorno y configuración específicos.

Para cambiar el color de la ventana en ImGui, puedes utilizar la función `ImGui::PushStyleColor()` antes de comenzar a dibujar la ventana. Aquí tienes un ejemplo básico de cómo puedes hacerlo:

```c
#include "imgui.h"
#include "imgui_impl_sdl.h"
#include "imgui_impl_opengl3.h"

int main()
{
    // Inicializar ImGui y crear una ventana
    ImGui::CreateContext();
    ImGuiIO& io = ImGui::GetIO(); (void)io;

    // Inicializar ImGui para SDL y OpenGL
    ImGui_ImplSDL2_InitForOpenGL(window, gl_context);
    ImGui_ImplOpenGL3_Init(glsl_version);

    // Loop principal
    while (!quit)
    {
        // Iniciar un nuevo frame de ImGui
        ImGui_ImplOpenGL3_NewFrame();
        ImGui_ImplSDL2_NewFrame(window);
        ImGui::NewFrame();

        // Cambiar el color de fondo de la ventana
        ImGui::PushStyleColor(ImGuiCol_WindowBg, ImVec4(0.2f, 0.2f, 0.2f, 1.0f)); // Color gris
oscuro

        // Comenzar a dibujar la ventana
        ImGui::Begin("Mi Ventana");

        // Contenido de la ventana

        // Finalizar la ventana
        ImGui::End();
```

```
        // Restaurar el color de fondo por defecto
        ImGui::PopStyleColor();

        // Renderizar los elementos de ImGui
        ImGui::Render();
        ImGui_ImplOpenGL3_RenderDrawData(ImGui::GetDrawData());

        // Actualizar la ventana y manejar eventos
        SDL_GL_SwapWindow(window);
        SDL_Event event;
        while (SDL_PollEvent(&event))
        {
            ImGui_ImplSDL2_ProcessEvent(&event);
            if (event.type == SDL_QUIT)
                quit = true;
        }
    }

    // Limpiar ImGui al finalizar
    ImGui_ImplOpenGL3_Shutdown();
    ImGui_ImplSDL2_Shutdown();
    ImGui::DestroyContext();

    return 0;
}
```

En este ejemplo, `ImGui::PushStyleColor()` se utiliza para cambiar el color de fondo de la ventana a un tono gris oscuro. Puedes ajustar los valores RGB en `ImVec4()` para personalizar el color según tus preferencias. Recuerda que este código es solo un ejemplo básico y puede necesitar modificaciones adicionales según tus necesidades específicas.

Para mantener las ventanas colapsadas en ImGui, puedes utilizar la función `ImGui::SetNextWindowCollapsed()` antes de comenzar a dibujar la ventana. Aquí tienes un ejemplo básico de cómo puedes hacerlo:

```c
#include "imgui.h"
#include "imgui_impl_sdl.h"
#include "imgui_impl_opengl3.h"

int main()
{
    // Inicializar ImGui y crear una ventana
    ImGui::CreateContext();
    ImGuiIO& io = ImGui::GetIO(); (void)io;

    // Inicializar ImGui para SDL y OpenGL
    ImGui_ImplSDL2_InitForOpenGL(window, gl_context);
    ImGui_ImplOpenGL3_Init(glsl_version);
```

```
    // Loop principal
    while (!quit)
    {
        // Iniciar un nuevo frame de ImGui
        ImGui_ImplOpenGL3_NewFrame();
        ImGui_ImplSDL2_NewFrame(window);
        ImGui::NewFrame();

        // Establecer que la ventana esté colapsada
        ImGui::SetNextWindowCollapsed(true, ImGuiCond_FirstUseEver);

        // Comenzar a dibujar la ventana
        ImGui::Begin("Mi Ventana");

        // Contenido de la ventana

        // Finalizar la ventana
        ImGui::End();

        // Renderizar los elementos de ImGui
        ImGui::Render();
        ImGui_ImplOpenGL3_RenderDrawData(ImGui::GetDrawData());

        // Actualizar la ventana y manejar eventos
        SDL_GL_SwapWindow(window);
        SDL_Event event;
        while (SDL_PollEvent(&event))
        {
            ImGui_ImplSDL2_ProcessEvent(&event);
            if (event.type == SDL_QUIT)
                quit = true;
        }
    }

    // Limpiar ImGui al finalizar
    ImGui_ImplOpenGL3_Shutdown();
    ImGui_ImplSDL2_Shutdown();
    ImGui::DestroyContext();

    return 0;
}
```

En este ejemplo, `ImGui::SetNextWindowCollapsed(true, ImGuiCond_FirstUseEver)` se utiliza para establecer que la ventana esté colapsada al inicio, pero solo la primera vez que se muestra. Puedes ajustar los parámetros según tus necesidades, como cambiar `true` a `false` si deseas que la ventana esté expandida por defecto.

Recuerda que este código es solo un ejemplo básico y puede necesitar modificaciones adicionales según tus necesidades específicas.

`ImGuiCond_FirstUseEver` es una constante utilizada en la función `ImGui::SetNextWindowCollapsed()` de la biblioteca ImGui. Esta constante se utiliza para especificar una condición de activación que indica que la configuración dada solo se aplicará la primera vez que se muestre la ventana.

En otras palabras, al utilizar `ImGuiCond_FirstUseEver`, estás indicando que la configuración de colapsar la ventana solo se aplicará la primera vez que se muestre esa ventana en particular. En las siguientes ocasiones en las que se muestre la ventana, esta configuración no se aplicará nuevamente, permitiendo que la ventana se mantenga en el estado en el que fue dejada por el usuario.

En resumen, `ImGuiCond_FirstUseEver` es una forma de controlar el comportamiento de ciertas configuraciones en ImGui para que se apliquen solo la primera vez que se cumple una condición específica, en este caso, al mostrar una ventana.

En la función `ImGui::SetNextWindowCollapsed()`, además de `ImGuiCond_FirstUseEver`, puedes utilizar otras constantes para controlar el comportamiento de la ventana. Aquí tienes algunas de las constantes que puedes utilizar junto con `SetNextWindowCollapsed()` en ImGui:

1. `ImGuiCond_Always`: Esta constante indica que la configuración dada siempre se aplicará, independientemente de si la ventana se ha mostrado anteriormente o no.

2. `ImGuiCond_Once`: Esta constante indica que la configuración dada se aplicará solo una vez, la primera vez que se cumpla la condición especificada.

3. `ImGuiCond_Appearing`: Esta constante indica que la configuración se aplicará solo cuando la ventana esté apareciendo, es decir, cuando se está mostrando por primera vez en la sesión actual.

Estas constantes te permiten controlar cuándo se aplican ciertas configuraciones a las ventanas en ImGui, brindándote flexibilidad para adaptar el comportamiento de las ventanas según tus necesidades específicas. Puedes experimentar con estas constantes para lograr el comportamiento deseado en tus interfaces de usuario creadas con ImGui.

Para ocultar una ventana en ImGui, puedes utilizar la función `ImGui::SetNextWindowCollapsed()` con el parámetro `collapsed` establecido en `true`. Esto colapsará la ventana y la ocultará visualmente. Aquí tienes un ejemplo básico de cómo puedes ocultar una ventana en ImGui:

```c
#include "imgui.h"
#include "imgui_impl_sdl.h"
#include "imgui_impl_opengl3.h"

int main()
{
    // Inicializar ImGui y crear una ventana
    ImGui::CreateContext();
    ImGuiIO& io = ImGui::GetIO(); (void)io;

    // Inicializar ImGui para SDL y OpenGL
    ImGui_ImplSDL2_InitForOpenGL(window, gl_context);
    ImGui_ImplOpenGL3_Init(glsl_version);

    bool showWindow = true; // Variable para controlar la visibilidad de la ventana
```

```
    // Loop principal
    while (!quit)
    {
        // Iniciar un nuevo frame de ImGui
        ImGui_ImplOpenGL3_NewFrame();
        ImGui_ImplSDL2_NewFrame(window);
        ImGui::NewFrame();

        // Establecer que la ventana esté colapsada para ocultarla
        ImGui::SetNextWindowCollapsed(!showWindow, ImGuiCond_Always);

        // Comenzar a dibujar la ventana
        ImGui::Begin("Mi Ventana", &showWindow);

        // Contenido de la ventana

        // Finalizar la ventana
        ImGui::End();

        // Renderizar los elementos de ImGui
        ImGui::Render();
        ImGui_ImplOpenGL3_RenderDrawData(ImGui::GetDrawData());

        // Actualizar la ventana y manejar eventos
        SDL_GL_SwapWindow(window);
        SDL_Event event;
        while (SDL_PollEvent(&event))
        {
            ImGui_ImplSDL2_ProcessEvent(&event);
            if (event.type == SDL_QUIT)
                quit = true;
        }
    }

    // Limpiar ImGui al finalizar
    ImGui_ImplOpenGL3_Shutdown();
    ImGui_ImplSDL2_Shutdown();
    ImGui::DestroyContext();

    return 0;
}
```

En este ejemplo, la variable `showWindow` se utiliza para controlar la visibilidad de la ventana. Al establecer `ImGui::SetNextWindowCollapsed(!showWindow, ImGuiCond_Always)`, la ventana se colapsará y se ocultará cuando `showWindow` sea `false`. Puedes cambiar el valor de `showWindow` según tus necesidades para mostrar u ocultar la ventana en función de alguna lógica específica de tu aplicación.

Para ocultar un widget específico en ImGui, puedes utilizar una combinación de lógica condicional y la función `ImGui::Begin()` para controlar cuándo se muestra o se oculta un widget en función de una condición. Aquí tienes un ejemplo básico de cómo puedes ocultar un widget en ImGui:

```c
#include "imgui.h"
#include "imgui_impl_sdl.h"
#include "imgui_impl_opengl3.h"

int main()
{
    // Inicializar ImGui y crear una ventana
    ImGui::CreateContext();
    ImGuiIO& io = ImGui::GetIO(); (void)io;

    // Inicializar ImGui para SDL y OpenGL
    ImGui_ImplSDL2_InitForOpenGL(window, gl_context);
    ImGui_ImplOpenGL3_Init(glsl_version);

    bool showWidget = true; // Variable para controlar la visibilidad del widget

    // Loop principal
    while (!quit)
    {
        // Iniciar un nuevo frame de ImGui
        ImGui_ImplOpenGL3_NewFrame();
        ImGui_ImplSDL2_NewFrame(window);
        ImGui::NewFrame();

        // Comenzar a dibujar la ventana
        ImGui::Begin("Mi Ventana");

        // Mostrar u ocultar el widget basado en la variable showWidget
        if (showWidget)
        {
            ImGui::Text("Este es un widget visible");
        }

        // Botón para alternar la visibilidad del widget
        if (ImGui::Button("Alternar Widget"))
        {
            showWidget = !showWidget;
        }

        // Finalizar la ventana
        ImGui::End();

        // Renderizar los elementos de ImGui
        ImGui::Render();
        ImGui_ImplOpenGL3_RenderDrawData(ImGui::GetDrawData());

        // Actualizar la ventana y manejar eventos
```

```
        SDL_GL_SwapWindow(window);
        SDL_Event event;
        while (SDL_PollEvent(&event))
        {
            ImGui_ImplSDL2_ProcessEvent(&event);
            if (event.type == SDL_QUIT)
                quit = true;
        }
    }

    // Limpiar ImGui al finalizar
    ImGui_ImplOpenGL3_Shutdown();
    ImGui_ImplSDL2_Shutdown();
    ImGui::DestroyContext();

    return 0;
}
```

En este ejemplo, el widget se muestra u oculta dependiendo del valor de la variable `showWidget`. Puedes cambiar la visibilidad del widget al hacer clic en el botón "Alternar Widget". Esta es una forma sencilla de controlar la visibilidad de un widget en ImGui basándote en alguna lógica específica de tu aplicación.